IMPROVED CRT ALGORITHM FOR CLASS POLYNOMIALS IN GENUS 2

DAMIEN ROBERT AND KRISTIN E. LAUTER

1. Introduction

Cryptographic solutions to provide privacy and security for sensitive transactions depend on using a mathematical group where the discrete logarithm problem is hard. For example, digital signature schemes or a Diffie-Hellman key exchange may be based on the difficulty of solving the discrete logarithm problem in the group of points on the Jacobian of a genus 2 curve. For this problem to be hard we must ensure that we can choose genus 2 curves over finite fields with an almost prime number of points on the Jacobian of the curve.

One approach to this problem is to construct curves with Jacobian of a given order using the (CM) method of Complex Multiplication. The CM method works by computing invariants of the curve and then reconstructing the curve using the Mestre-Cardona-Quer [23] algorithm. Invariants are computed by constructing their minimal polynomials, called Igusa class polynomials. Computing the invariants is computationally intensive, and there are three known methods for constructing Igusa class polynomials:

(1) the complex analytic method [26, 29, 30, 27];
(2) the Chinese Remainder Theorem method (CRT) [12, 14, 5]; and
(3) the p-adic lifting method [15, 7, 8].

Although the CRT method in genus 2 is currently still by far the slowest of these three methods as measured on small examples which have been computed to date, there is some hope that it may be asymptotically competitive with the other methods based on the history of the evolution of these three methods in genus 1. Asymptotically, and also for space constraint reasons, the (Explicit) CRT method now holds the record in genus 1 for the size of the largest examples computed via that method [28, 13]. In this paper, we propose numerous improvements to the CRT method for computing genus 2 curves, paralleling improvements made by Sutherland [28] to the CRT method in genus 1.

The CRT method works by computing class polynomials modulo many small primes, and then reconstructing the polynomial with rational coefficients (or modulo a much larger prime number) via the Chinese Remainder Theorem (*resp.* the Explicit CRT). The CRT method for computing class polynomials in genus 2 was proposed by [12], with sufficient conditions on the CRT primes to ensure correctness and including an algorithm for computing endomorphism rings for ordinary Jacobians of genus 2 curves which generalized Kohel's algorithm for genus 1 curves. For each small CRT prime p, the algorithm loops through all p^3 possible triples of Igusa

Date: September 13, 2010.

2000 *Mathematics Subject Classification.* Primary 11G20; Secondary .

invariants of curves, reconstructing the curve and testing for each curve whether it is in the desired isogeny class and whether its endomorphism ring is maximal. The algorithm for computing endomorphism rings from [12] was replaced by a much more efficient probabilistic algorithm in [14], where a number of examples were given for running times of the computations modulo small CRT primes. [5] introduced the idea of using computable $(3, 3)$-isogenies to find other curves in the isogeny class once an initial curve was found, but still searched until finding a curve with maximal endomorphism ring. Another improvement described in [5] was a method to construct other maximal curves using $(3, 3)$ isogenies once a first maximal curve is found.

In this paper we present a generalization to genus 2 of the probabilistic Algorithm 1 in Sutherland [28]. The improvement over the algorithm presented in [5] for the genus 2 case, is that, here we do not need to find a curve in the isogeny class with endomorphism ring which is the maximal order: rather we present a probabilistic algorithm for "going up" to a *maximal* curve (a curve with maximal endomorphism ring), once we find *any* curve in the right isogeny class. Then we use the structure of the Shimura class group and the computation of (ℓ, ℓ)-isogenies to compute all isogenous maximal curves from an initial one. Although we cannot prove that the "going up" algorithm succeeds with any fixed probability, it works well in practice, and heuristically it improves the running time of the genus 2 CRT method from p^3 per prime p to $p^{\frac{3}{2}}$ per prime p.

Let K denote a primitive quartic CM field, and Φ a CM type. Let K_Φ denote the reflex CM field, and K^+ the totally real subfield of K. For a field K, let $\mathcal{O}_K$ be the ring of integers of K. Let TN_Φ the typenorm associated to the CM-type Φ. Informally, the algorithm is as follows, and individual steps will be explained in subsequent sections:

Algorithm 1.
Input: A primitive quartic cyclic CM field K with a CM type Φ, a collection of CRT primes P_K for K.
Output: Igusa Class Polynomials $H_i(x)$, $i = 1, 2, 3$ in $\mathbb{Q}[x]$ or modulo a prime q.

- *Loop through CRT primes $p \in P_K$:*
    1. *Enumerate hyperelliptic curves C of genus 2 over $\mathbb{F}_p$ until a curve in the right $\mathbb{F}_p$-isogeny class (up to a quadratic twist) is found.*
    2. *Try to go up to a maximal curve from C, if it fails go back to Step 1.*
    3. *From a maximal curve C, compute all other maximal curves.*
    4. *Reconstruct the class polynomials $H_i(x)$ modulo p from the Igusa invariants of the set of maximal curves.*
- *Recover $H_i(x)$, $i = 1, 2, 3$ in $\mathbb{Q}[x]$ or modulo q using the (Explicit) CRT method once we have $H_i(x)$ modulo p for enough primes p.*

For the dihedral case, one new aspect of our algorithm is that we extend to the CRT setting the idea of computing the class polynomials associated to only one fixed CM type Φ for K [27, Section III.3]. When K is cyclic, this makes no difference, since all isomorphism classes of abelian surfaces with CM by K arise from one CM type; but when K is dihedral, two CM types are needed to find all isomorphism classes of CM abelian surfaces. All three previous versions of the CRT algorithm [12, 14, 5] compute the class polynomials classifying all abelian varieties with CM by $\mathcal{O}_K$ (with either of the two possible CM types in the dihedral case). The advantage of our approach is that it computes only a factor of half the degree of the whole class polynomial. The drawback of this approach is that in the dihedral case, each factor of the class polynomials is defined over $\mathcal{O}_{K_\Phi^+}$ rather than over $\mathbb{Z}$. So once we compute the class polynomials modulo $\mathfrak{p}$ as polynomials in $\mathcal{O}_{K_\Phi^+}/\mathfrak{p}$, the CRT step must be performed in $\mathcal{O}_{K_\Phi^+}$.

A CRT prime $\mathfrak{p} \subset \mathcal{O}_{K_\Phi^+}$ is a prime such that all abelian varieties over $\mathbb{C}$ with CM by $(\mathcal{O}_K, \Phi)$ have good reduction modulo $\mathfrak{p}$. By [25, Section III.13], $\mathfrak{p}$ is a CRT prime for the CM type Φ if and only if there exists an unramified prime $\mathfrak{q}$ in $\mathcal{O}_{K_\Phi}$ of degree 1 above $\mathfrak{p}$ of principal type norm (π) (in particular this implies that $\mathfrak{q}$ is totally split in the class field corresponding to the abelian varieties with CM by $(\mathcal{O}_K, \Phi)$). Moreover, by a theorem of Tate, the isogeny class of these abelian varieties reduced modulo $\mathfrak{p}$ (by [16, Section 3] they have good reduction) is determined by the characteristic polynomial of $\pm\pi$ (here we assume that $\mathcal{O}_K^* = \pm 1$). For efficiency reasons, we will work with CRT primes $\mathfrak{p}$ that are unramified of degree one over $p = \mathfrak{p} \cap \mathbb{Z}$. By [16], the reduction to $\mathbb{F}_p$ of the abelian varieties with CM by $(\mathcal{O}_K, \Phi)$ will then be ordinary. We then make the slight abuse of notation of calling p a CRT prime when there is a CRT prime $\mathfrak{p}$ above it. Note another advantage of restricting to one CM type: to use p for both CM types, p needs to split completely into $p = \mathfrak{p}_1\mathfrak{p}_2$ such that both $\mathfrak{p}_1$ and $\mathfrak{p}_2$ are CRT primes, and there are fewer p which satisfy this stronger requirement.

In addition to the two main contributions of the paper, the “going up” algorithm to find maximal curves, and an improvement to the algorithm to compute maximal curves from maximal curves, we also give improvements to every step of the CRT algorithm. Here we give a brief outline of the paper and a summary of those improvements.

Step 2 (the “going-up” part) of the algorithm is explained in Section 3. We first explain in Section 2 how to compute if a curve is maximal, since this is used in the going-up algorithm. We present some significant improvements over the algorithm from [14]. Step 3 (finding all other maximal curves from one maximal curve) is explained in Section 4.

As for Step 4, once all maximal abelian varieties with CM by K are found for a given prime p, it is easy to compute the associated class polynomials modulo p. The class polynomials depend on the choice of Igusa invariants, and we use the invariants recommended in [27, Appendix 3] which give smaller coefficients than those used in [29, 30, 16]. For the dihedral case the class polynomials must be reconstructed over $\mathcal{O}_{K_\Phi^+}$, and we give more details about this step in Section 5.

Section 6 gives a complexity analysis, and explains how each improvement affects the final complexity. The final complexity bound, while still not quasilinear, is a significant improvement compared to [5]. Finally, examples demonstrating significantly improved running times are given in Section 7.

2. Checking if the endomorphism ring is maximal

We recall the algorithm described in [12], and describe some improvements. The ideas for computing the endomorphism ring will be used in the going up phase.

2.1. **The algorithm of Eisenträger, Freeman and Lauter.** Let $A/\mathbb{F}_p$ be an ordinary abelian variety of dimension 2 with CM by K. Let $O = \mathrm{End}(A)$. We know that $\mathbb{Z}[\pi] \subset \mathbb{Z}[\pi, \overline{\pi}] \subset O \subset O_K$. We want to check if $O = O_K$. First, the Chinese Remainder Theorem gives us the following proposition:

Proposition 2. *Let O be an order in $\mathcal{O}_K$. Let $\{1, \alpha_1, \alpha_2, \alpha_3\}$ be a basis of $\mathcal{O}_K$ as a $\mathbb{Z}$-module. Write $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]] = \prod \ell_i^{e_{\ell_i}}$. If $\frac{[\mathcal{O}_K : \mathbb{Z}[\pi,\overline{\pi}]]}{\ell_i^{e_{\ell_i}}} \alpha_j \in O$ for $j = 1, 2, 3$, and all ℓ_i dividing the index, then $O = \mathcal{O}_K$.*

We are then reduced to the following problem: for $\gamma \in \mathcal{O}_K$ such that $\ell^e \gamma \in \mathbb{Z}[\pi, \overline{\pi}]$, check if $\gamma \in O$. To simplify the notation,we will often drop the subscript on e_ℓ when it is obvious to which ℓ we are referring to. We have ([14]):

Proposition 3. *Let $O = \mathrm{End}(A)$ and $\gamma \in \mathcal{O}_K$. There exists an integer polynomial P_γ such that $\ell^e p \gamma = P_\gamma(\pi)$. Then γ is in O if and only if $P_\gamma(\pi) = 0$ on $A[\ell^e]$.*

Proof. First note that $[\mathbb{Z}[\pi, \overline{\pi}] : \mathbb{Z}[\pi]] = p$, (see [14]), so that $\ell^e p \gamma \in \mathbb{Z}[\pi]$, which means we can write: $\ell^e p \gamma = P_\gamma(\pi)$. Second, since we are dealing with ordinary abelian varieties over $\mathbb{F}_p$, we have $p \not| [O_K : \mathbb{Z}[\pi, \overline{\pi}]]$ [14, Proposition 3.7], so that $\gamma \in O \Leftrightarrow p\gamma \in O$. Lastly, by the universal property of isogenies, we have that $P_\gamma(\pi) = 0$ on $A[\ell^e]$ if and only if $p\gamma \in O$ (see [12]). Summing up, we only need to check that $P_\gamma(\pi) = 0$ on $A[\ell^e]$ to check that $\gamma \in O$. □

Remark 4. *Since most of the curves in the isogeny class are not maximal, it is more efficient to check P_γ on $A[\ell]$, $A[\ell^2]$, $\ldots$, rather than directly on $A[\ell^e]$.*

2.2. Computing the ℓ^e-torsion.

The main cost of the preceding algorithm is to compute a basis of the ℓ^e-torsion groups. The cost of such a computation depends on the degree of the extension where the ℓ^e-torsion points are defined. We have:

Lemma 5. *Let d be the degree such that the ℓ-torsion points of A are defined over $\mathbb{F}_{p^d}$. Then $d \leq \ell^4 - 1$. Furthermore, the ℓ^e-torsion is all defined over $\mathbb{F}_q$ with $q = p^{d\ell^{e-1}}$.*

Proof. Let χ_π be the characteristic polynomial of Frobenius, π. Then d is the (multiplicative) order of X in the ring $\mathbb{F}_\ell[X]/\chi_\pi(X)$, so $d \leq \ell^4 - 1$. The second assertion follows from [14, Section 6]. □

Remark 6. [14, Proposition 6.2] *gives a better bound for maximal abelian surfaces: in that case we have $d \leq \ell^3$, and if ℓ is completely split in $\mathcal{O}_K$, we have $d \mid \ell - 1$.*

We will use the following algorithm to compute points uniformly in an ℓ-primary group containing $A[\ell^e]$:

Algorithm 7. *Precomputation:*

a) Let d be the (multiplicative) order of X in the ring $\mathbb{F}_\ell[X]/\chi_\pi(X)$ and set $d_e = d\ell^{e-1}$.

b) Compute $\chi_{\pi^{d_e}}$ as the resultant in X of $\chi_\pi(Y)$ and $Y^{d_e} - X$, and write $\#A(\mathbb{F}_p^{d_e}) = \chi_{\pi^{d_e}}(1) = \ell^e \gamma$ with γ prime to ℓ.

Compute uniform random points in the ℓ-primary component $A(\mathbb{F}_{p^{d_e}})[\ell^\infty]$:

1. *Take a random point P (uniformly) in $A(\mathbb{F}_{p^{d_e}})$;*
2. *return γP.*

Algorithm 4.3 of [14] takes random (uniform) points P in $A(\mathbb{F}_{p^{d_e}})[\ell^\infty]$ to get random points in $A(\mathbb{F}_{p^{d_e}})[\ell^e]$ by looking at the smallest k such that $\ell^k.P$ is an ℓ^e-torsion point, then generates enough such random points so that the probability that they generate the full ℓ^e-torsion is sufficiently high and then tests P_γ on these points of ℓ^e-torsion. The algorithm computes how many points are needed so that the probability of generating the full ℓ^e-torsion is greater than $1-\epsilon$ for some $\epsilon > 0$, so the result is not guaranteed (i.e. it is a "Monte-Carlo" algorithm), which is very inconvenient in our setting since we need to test a lot of curves across different CRT primes p.

To ensure correctness we can check that the subgroup generated by the points obtained is of cardinality ℓ^{2ge}, but this is costly. A more efficient way is as follows: $\{P_1, \ldots, P_{2g}\}$ is a basis of the ℓ^e-torsion if and only if $\{\ell^{e-1}P_1, \ldots, \ell^{e-1}P_{2g}\}$ is a basis of the ℓ-torsion. But that can be easily checked by computing the $g(2g-1)$ Weil pairings $e_\ell(\ell^{e-1}P_i, \ell^{e-1}P_j)$ for $i < j$ and testing if we get an invertible matrix. Since Weil pairings can be computed in $O(\log(\ell))$, this is much faster. This is our first improvement, yielding a "Las-Vegas" algorithm.

The second drawback of the approach of [14] is that, although the random points in $A(\mathbb{F}_{p^{d_e}})[\ell^\infty]$ are uniform, this is not always the case for the random points in $A(\mathbb{F}_{p^{d_e}})[\ell^e]$. To have a high probability of generating the full ℓ-torsion then requires taking many random points in $A(\mathbb{F}_{p^{d_e}})[\ell^\infty]$: if $A(\mathbb{F}_{p^{d_e}})[\ell^\infty] = \ell^s$, the algorithm requires $\ell^{s-4e}(-\log(\epsilon))^{1/2}$ random points to succeed with probability greater than $1-\epsilon$. Since generating these points is the most costly part of the algorithm it is best to minimize the number of random points required. Our second improvement is to use an algorithm, implemented in AVIsogenies, due to Couveignes [10] to get uniform random points in $A(\mathbb{F}_{p^{d_e}})[\ell^e]$. Since the full algorithm is described in more detail in [3], we only give an example to illustrate it here.

Suppose that G is an ℓ-primary group generated by a point P of order ℓ^2 and a point Q of order ℓ. Assume that the first random point chosen is $P = R_1$, which gives an ℓ-torsion point $T_1 = \ell P$. The second random point R_2 chosen will be of the form $\alpha P + \beta Q$. In most cases, $\alpha \neq 0$, so the corresponding new ℓ-torsion point is $T_2 = \alpha \ell P$, a multiple of T_1. However we can correct R_2 by the corresponding multiple: compute $R_2' = R_2 - \alpha R_1 = \beta Q$. Thus R_2' gives the rest of the ℓ-torsion except if $\beta = 0$. In our setting we can use the Weil pairing to express a new ℓ-torsion point in terms of the generating set already constructed (except when we have an isotropic group, in this case we have to compute the ℓ^2 multiples), and we only need $O(1)$ random points to find a basis. Neglecting logarithmic factors in ℓ the cost of finding a basis of the ℓ^e-torsion is then $\tilde{O}((d_e + \ell^2)\log p)$.

2.3. **Reducing the degree.** The complexity of finding the basis is closely related to the degree of the extension d_e. Let d_0 be the minimal integer such that $(\pi^{d_0} - 1) \in \ell\mathcal{O}_K$. Then $d_0 \mid d$, and as remarked in [14], since we only need to check if $O = \mathcal{O}_K$, we can first check that $\frac{\pi^{d_0}-1}{\ell}$ lies in O. In other words, we can check that the ℓ-torsion points of A are defined over $\mathbb{F}_{p^{d_0}}$ rather than over $\mathbb{F}_{p^d}$. If this is the case, the ℓ^e-torsion points are then defined over an extension of degree $d_0\ell^{e-1}$ of $\mathbb{F}_p$, which allows working with smaller extensions.

Another improvement we implemented to reduce the degree is to use twists. Let d_0' be the minimal integer such that $((-\pi)^{d_0'} - 1) \in \ell\mathcal{O}_K$. Then we have three possibilities $d_0' = d_0$, $d_0' = 2d_0$, $d_0 = 2d_0'$. In the latter case, it is better to replace A by its twist, since the Frobenius of the twist is represented by $-\pi$, and we can compute the points of ℓ^e-torsion by working over extensions of half the degree.

Example 8. *Let $H : y^2 = 80x^6 + 51x^5 + 49x^4 + 3x^3 + 34x^2 + 40x + 12$ be a hyperelliptic curve of genus 2 defined over $\mathbb{F}_{139}$ and J the Jacobian of H. We have* $\mathrm{End}(J) \otimes \mathbb{Q} \cong \mathbb{Q}(i\sqrt{13 + 2\sqrt{29}})$ *and we want to check if* $\mathrm{End}(J)$ *is maximal. In* *this example, we compute* $[O_K : \mathbb{Z}[\pi, \overline{\pi}]] = 3^5$*, so we need to compute* $J[3^5]$*, which lives over an extension of degree* 81*. If we had checked the endomorphism ring of the Jacobian of the twist of* H*, we would have needed to work over an extension of degree* 162*.*

2.4. **Reducing the number of endomorphisms to test.** One last improvement to the algorithm of [14] is to use the fact that $\mathrm{End}(A)$ is an order. So if we know that $\gamma \in O$, then we know that the ring $\mathbb{Z}[\pi, \overline{\pi}, \gamma] \subset O$. As an example, if in terms of our basis we have $\alpha_3 = \alpha_2 \alpha_1 \mod O_K^*$, then we only have to check that α_2 and α_1 are in O (and since our algorithm works locally at ℓ, we only need that this relation is true locally for that ℓ). We use this idea as follows: suppose that we have checked that $\{\gamma_i, i = \{1, \dots, k\}\}$ are endomorphisms lying in O, and we want to check if $\gamma \in O$. Let N_1 be the order of γ in $O_K/\mathbb{Z}[\pi, \overline{\pi}, \gamma_i : i = 1, \dots, k]$, and N_2 be the order of γ in $O_K/\mathbb{Z}[\pi, \overline{\pi}]$. If we write $N_2 = \prod \ell_i^{e_{\ell_i}}$, we only have to check that $N_2/\ell^{e_\ell}\gamma \in O$ for $\ell \mid N_1$. In fact, if the valuation of N_1 at ℓ is e'_ℓ, then we would only need to check that $N_1/\ell^{e'_\ell}\gamma \in O$, which means testing if $N_1\gamma = 0$ on the $\ell^{e'_\ell}$-torsion, where $N_1\gamma$ is a polynomial in π, $\overline{\pi}$, and the γ_i $(i = 1, \dots, k)$. We write this polynomial as $\frac{N_1}{pN_2}$ times a polynomial in π, so that we still need to compute the ℓ^{e_ℓ}-torsion.

Example 9. *Let* $H : y^2 = 10x^6 + 57x^5 + 18x^4 + 11x^3 + 38x^2 + 12x + 31$ *be a genus* 2 *curve defined over* $\mathbb{F}_{59}$ *and* J *the Jacobian of* H*. We have* $\mathrm{End}(J) \otimes \mathbb{Q} = \mathbb{Q}(i\sqrt{29 + 2\sqrt{29}})$ *and we want to check if* $\mathrm{End}(J) = \mathcal{O}_K$*.* $\mathcal{O}_K$ *is generated as a* $\mathbb{Z}$*-module by* $1, \alpha, \beta, \gamma$*, where* α *is of index* 2 *in* $\mathcal{O}_K/\mathbb{Z}[\pi, \overline{\pi}]$*,* β *of index* 4 *and* γ *of index* 40*. The algorithm from* [14] *would check* $J[2^3]$ *and* $J[5]$*. But* $(\mathcal{O}_K)_2 = \mathbb{Z}_2[\pi, \overline{\pi}, \alpha]$*, so we only need to check* $J[2]$ *and* $J[5]$*.*

2.5. **The algorithm.** Incorporating all these improvements yields the following algorithm:

Algorithm 10. *Checking that* $\mathrm{End}\, A$ *is maximal.*

```
Input: An ordinary abelian surface $A/\mathbb{F}_p$ with CM by $K$.
Output: The boolean statement: $\mathrm{End}\, A = O_K$.
(1) Choose a basis $\{1, \alpha_1, \alpha_2, \alpha_3\}$ of $O_K$ and a basis $\{1, \beta_1, \beta_2, \beta_3\}$ of $\mathbb{Z}[\pi]$ such
    that $\beta_1 = c_1\alpha_1$, $\beta_2 = c_2\alpha_2$, $\beta_3 = c_3\alpha_3$ and $c_1, c_2, c_3 \in \mathbb{Z}$ with $c_1 \mid c_2 \mid c_3$.
(2) (Checking where the $\ell$-torsion lives.) For each $\ell \mid [O_K : \mathbb{Z}[\pi, \overline{\pi}]]$ do
    (a) Let $d_\ell$ be the smallest integer such that $\pi^{d_\ell} - 1 \in \ell O_K$, and $d'_\ell$ be the
        smallest integer such that $(-\pi)^{d'_\ell} - 1 \in \ell O_K$. If $d'_\ell < d_\ell$, switch to the
        quadratic twist.
    (b) Compute a basis of $A[\ell](\mathbb{F}_{p^{d_\ell}})$ using the algorithm from [3].
    (c) If this basis if of cardinality (strictly) less than four, return false.
    (d) (Checking the generators of $O_K$.) For $i = 1, 2, 3$ do
        (i) Let $N_1$ be the order of $\alpha_i$ in $O_K/\mathbb{Z}[\pi, \overline{\pi}, \alpha_j : j < i]$ and $N_2$ the
            order of $\alpha_i$ in $O_K/\mathbb{Z}[\pi, \overline{\pi}]$.
        (ii) If $\ell \mid N_1$, let $e_i$ be the $\ell$-valuation of $N_2$ and write $pN_2\alpha_i$ as a
            polynomial $P_i(\pi)$.
        (iii) Compute a basis of $A(\mathbb{F}_{p^{d_\ell \ell^{e_i - 1}}})[\ell^{e_i}]$.
        (iv) If $P_i(\pi) \neq 0$ on this basis, return false.
(3) Return true.
```

2.6. **Complexity.** The complexity is measured in terms of operations in the base field, $\mathbb{F}_p$, neglecting logarithmic factors $\log(p)$. Since the index $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$ is bounded by a polynomial in p by [14, Proposition 6.2], evaluating the polynomials $P_i(\pi)$ (of degrees at most 3) is done in logarithmic time. The most expensive part of the algorithm is then the computation of $A[\ell^e]$, for the various ℓ dividing the index $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$ where e is at most the ℓ-valuation of the index. According to Lemma 5 and Remark 6, the ℓ^e-torsion points live in an extension of degree at most $d = \ell^{e+3}$. Since $\#A(\mathbb{F}_{p^d}) = p^{2d(1+\epsilon)}$, computing a random point in $A(\mathbb{F}_{p^d})[\ell^e]$ takes $\tilde{O}(d^2)$. Correcting this random point requires some pairing computations, and costs at most $\tilde{O}(\ell^2)$ (in case the first points give an isotropic group). Since we need $O(1)$ such random points, the global cost is given by the following proposition (we will only need a very rough bound for the complexity analysis in Section 6):

Proposition 11. *Let $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]] = \prod \ell_i^{e_i}$ be the decomposition of the index into prime factors. Then checking if an abelian surface in the isogeny class is maximal can be done in time $\sum \tilde{O}(\ell_i^{2e_i+6})$.*

Remark 12. *One can compare to* [14, Proposition 4.6] *to see the speedup we gain in the endomorphism ring computation. We note that our method is exponential in the discriminant, while in* [4] *one can find a subexponential algorithm to compute the endomorphism ring of an ordinary abelian surface. In ongoing work with Gaetan Bisson, we have developed a method that combines the going up algorithm of the next section with his endomorphism ring algorithm. Since we still need to take ℓ-isogenies for $\ell \mid [\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$ in the going up step, this approach is mainly interesting when the index is divisible by a power of a prime.*

3. Going up

"Going up" is the process of finding genus 2 curves with maximal endomorphism ring by moving from *any* curve in the isogeny class via isogenies to a maximal one. This is not always possible and we will explain some obstructions. One difficulty was already illustrated in [5, Example 8.3], where it was shown that there can be cycles in the isogeny graph involving some non-maximal curves. Clearly, when trying to "go up", the algorithm should avoid making cycles in the graph, and we propose one method to avoid that. Further difficulties arise from the fact that the graph of rational (ℓ, ℓ) isogenies can be disconnected and there can be some nodes with no rational (ℓ, ℓ)-isogenies. This is an important caveat, as this means that our method for "going-up" will not always succeed, so we only have a probabilistic algorithm, and we cannot currently estimate the probability of failure.

As noted in [14], for the type of fields we can deal with via the CRT method, the cost of going through p^3 Jacobians is dominant compared to checking if the endomorphism ring is maximal (this imbalance is magnified in our case due to our faster algorithm to compute the ℓ^e-torsion). In our algorithm, we try to find a random curve in the isogeny class, and we try to select p so that the probability of finding a curve in the right isogeny class is of magnitude $p^{3/2}$. In practice, finding one such curve is still the dominant aspect, which explains why we can afford to spend a lot of effort on "going up" from this curve.

The algorithm we propose for "going up" is made possible by the techniques developed in [22, 9, 24] for computing rational (ℓ, ℓ)-isogenies between abelian varieties over finite fields. If A is an ordinary abelian variety with CM by K, then for each ℓ dividing the index $[O_K : \mathbb{Z}[\pi, \overline{\pi}]]$, we try to find an (ℓ, ℓ)-isogeny path starting from A and going to A' such that $(\mathcal{O}_K)_\ell = \operatorname{End}(A')_\ell$. If this is possible, we let $A = A'$ in the next step (going to the next ℓ). A rather inefficient method for finding A' would be to use the algorithm for computing endomorphism rings which was detailed in the preceding section (modified to handle the case of non maximal orders), compute the endomorphism ring of $\operatorname{End}(A)$ and the (ℓ, ℓ)-isogenous varieties A', and keep A' if its endomorphism ring is bigger than the one of A. In this section we will describe a more efficient algorithm, that combines the endomorphism ring checks of the preceding section with a going up phase. Since we are working locally in ℓ, we may as well suppose that we are working over $\mathbb{Z}_\ell$.

3.1. **Going up for one endomorphism.**

In this section, we suppose that we have an element $\alpha' \in \mathcal{O}_K$ such that $\gamma \ell^e \alpha' \in \mathbb{Z}[\pi]$ with γ prime to ℓ. Starting from an abelian variety A in the isogeny class, we want to find an abelian variety A' such that $\frac{\alpha}{\ell^e} \in \operatorname{End}(A')$ (or equivalently that $\alpha' \in \operatorname{End}(A')$ locally at ℓ).

We saw in Section 2 that $\frac{\alpha}{\ell^e}$ is in the endomorphism ring of A iff $\alpha(A[\ell^e]) = 0$, and we know how to compute this subgroup. More generally, we let $N = \#\alpha(A[\ell^e])$. We think of N as a way to measure the "obstruction" to "$\frac{\alpha}{\ell^e} \in \operatorname{End}(A)$". Our algorithm is as follows: for each (ℓ, ℓ)-isogenous A', if $N' = \#\alpha(A'[\ell^e])$, then we replace A by A' if $N' < N$. We iterate this process until $N = 1$, in which case we have succeeded, or until we are stuck, in which case we try to find a new random abelian variety in the right isogeny class.

Rather than computing directly the obstruction $N = \#\alpha(A[\ell^e])$, we can compute the partial obstructions $N^\epsilon = \#\alpha(A[\ell^\epsilon])$ for $\epsilon \leq e$. Starting from $\epsilon = 1$, we take isogenies until we find an abelian variety A with $N^\epsilon = 1$, which means that $\frac{\alpha}{\ell^\epsilon} \in \operatorname{End}(A)$. We will now try to take isogenies to reduce the obstruction of higher degree $N^{\epsilon+1}$. Let $k = \alpha(A[\ell^{\epsilon+1}]) \subset A[\ell]$. The following lemma helps us select the isogeny we are looking for:

Lemma 13. *Let A' be an abelian variety isogenous to A, such that $\#\alpha(A'[\ell^{\epsilon+1}]) < \#\alpha(A[\ell^{\epsilon+1}])$. Then the kernel of the isogeny $A \to A'$ intersects non trivially with $k = \alpha(A[\ell^{\epsilon+1}])$.*

Proof. Let $f : A \to A'$ be a rational isogeny between A and A'. Then since α is a polynomial in the Frobenius, we have $\alpha \circ f = f \circ \alpha$. In particular, f maps $\alpha(A[\ell^{\epsilon+1}])$ to $\alpha(A'[\ell^{\epsilon+1}])$. If $\#\alpha(A'[\ell^{\epsilon+1}]) < \#\alpha(A[\ell^{\epsilon+1}])$ then there exists $x \in \operatorname{Ker} f \cap \alpha(A[\ell^{\epsilon+1}])$. □

This gives the following algorithm:

Algorithm 14. *Going up for one endomorphism $\frac{\alpha}{\ell^\epsilon}$.*

```
Input: An ordinary abelian variety $A/\mathbb{F}_p$ with CM by $K$.
Output: An abelian variety $A'/\mathbb{F}_p$ such that $\frac{\alpha}{\ell^\epsilon} \in \operatorname{End} A$ or fail.
(1) Set $\epsilon = 1$.
(2) Compute $N^\epsilon = \#\alpha(A[\ell^\epsilon])$.
(3) If $N = 1$, then if $\epsilon = e$ return $A$. Otherwise ($\epsilon < e$) set $\epsilon := \epsilon + 1$, and go
    back to Step 2.
(4) Else ($N > 1$) let $\mathcal{L}$ be the list of all rational maximal isotropic subgroups of
    $A[\ell]$ which intersect non trivially with $\alpha(A[\ell^\epsilon])$. For $k \in \mathcal{L}$ do
    (a) Compute $A' = A/k$. Let $N' = \#\alpha(A'[\ell^\epsilon])$. If $N' < N$, set $A = A'$ and
        go back to Step 2.
```

(5) *(We are stuck) Return* fail.

Remark 15. *As in Section 2 we let d_0 be the minimal integer such that $(\pi^{d_0}-1) \in \ell O_K$ and d the minimal integer such that $(\pi^d - 1) \in \ell\mathbb{Z}[\pi]$. Then the ℓ^ϵ-torsion points of A are defined over an extension of degree $d\ell^{\epsilon-1}$. If moreover $\frac{\pi^{d_0}-1}{\ell} \in \operatorname{End}(A)$ they are actually defined over an extension of degree $d_0\ell^{\epsilon-1}$.*

Therefore when we try to go up globally for all endomorphisms α, the first step is to try to go up for the endomorphism $\frac{\pi^{d_0}-1}{\ell}$. During the algorithm, the obstruction N is given by the size of the kernel of $\pi^{d_0}-1$, whose rank is $2g$ minus the rank of the ℓ-torsion points defined over $\mathbb{F}_{p^{d_0}}$. So we compute the size of a basis of $A[\ell](\mathbb{F}_{p^{d_0}})$ and take isogenies, where this size increases until we find the full rank.

3.2. **Going up globally.** Let $\{1, \frac{\alpha_i}{\ell^{e_i}}\}$ $(i = 1, 2, 3)$ be generators for the maximal order $(\mathcal{O}_K)_\ell$ over the subring $\mathbb{Z}_\ell[\pi, \overline{\pi}]$, where $\alpha_i \in \mathbb{Z}_\ell[\pi, \overline{\pi}]$. Starting from an abelian variety A in the isogeny class, we want to find an abelian variety which is maximal at ℓ.

We could apply Algorithm 14 for each $\frac{\alpha_i}{\ell^{e_i}}$, but it does not guarantee that the endomorphisms already defined on A stay defined during the process, so we would observe loops on non maximal abelian varieties with this method. Moreover we want to reuse the computations of $A[\ell^\epsilon]$ which are the expensive part of the process.

If $N_i = \#\alpha_i(A[\ell^{d_i}])$ for $i = 1, 2, 3$ is the obstruction corresponding to α_i, we define N to be the global obstruction $N = \sum N_i$. We can then adapt the same method: for each (ℓ,ℓ)-isogenous A', if $N'_i = \#\alpha_i(A'[\ell^{d_i}])$, then we replace A by A' if $\sum N'_i < \sum N_i$. We iterate this process until all the $N_i = 1$, in which case we go to the next ℓ, or until we are stuck, in which case we try to find a new random abelian variety in the right isogeny class.

As before, if $e = \max(e_1, e_2, e_3)$ we first compute $A[\ell^\epsilon]$ and the partial obstructions $N_i^\epsilon = \#A[\ell^{\min(\epsilon, e_i)}]$ $(i = 1, 2, 3)$. We do the same for the (ℓ,ℓ)-isogenous abelian varieties, and switch to the new one if $\sum N_i^\epsilon$ decreases (strictly). This allows working with smaller torsion in the beginning steps.

The level, ϵ, of the individual obstruction we are working on depends on the endomorphism considered, so if we get stuck on level ϵ, we may have to look at level $\epsilon+1$ even if not all endomorphisms $\frac{\alpha_i}{\ell^\epsilon}$ are defined yet. For instance (in this example we suppose we only deal with two generators) there are cases where $N_1^\epsilon = 1$, $N_2^\epsilon \neq 1$ and ${N'}_1^\epsilon = 1$, ${N'}_2^\epsilon = N_2^\epsilon$ for all (ℓ,ℓ)-isogenous abelian varieties A', so we are stuck on level ϵ. However we can still find an isogenous A' such that ${N'}_1^{\epsilon+1} < N_1^{\epsilon+1}$.

Finally, as in Remark 15, we first try to go up in a way that increases the size of $A(\mathbb{F}_{p^{d_0}})[\ell]$. If we are unlucky and get stuck, we switch to the computation of the full ℓ-torsion over $\overline{\mathbb{F}}_p$. This method allows working over the smallest extension to compute $A[\ell^e]$ as soon as possible.

A summary of the algorithm with the notation from above is given below:

Algorithm 16. *Going up.*

Input: *An ordinary abelian surface $A/\mathbb{F}_p$ with CM by K.*
Output: *An abelian variety $A'/\mathbb{F}_p$ with* $\operatorname{End} A = \mathcal{O}_K$ *(locally at ℓ) or* fail.

(1) *(Special case for the endomorphism $\frac{\pi^{d_0}-1}{\ell}$) Compute a basis B of $A(\mathbb{F}_{p^{d_0}})[\ell]$. If $\#B < 2g$, compute a basis B' of $A'(\mathbb{F}_{p^{d_0}})[\ell]$ for each (ℓ,ℓ)-isogenous abelian variety A'. If $\#B' > \#B$, restart the algorithm with $A' = A$. If $\#B = 4$ or we get stuck, go to the next step.*

(2) *Set $\epsilon = 1$.*
(3) *Compute*[1] $N_i^\epsilon = \#\alpha_i(A[\ell^{\min(\epsilon,e_i)}])$ *for $i = 1, 2, 3$.*
(4) *If $\{N_i : i = 1, 2, 3\} = \{1\}$ then if $\epsilon = \max(e_i : i = 1, 2, 3)$ return A. Else set $\epsilon := \epsilon + 1$ and go back to Step 3.*
(5) *Else let $\mathcal{L}$ be the list of all rational maximal isotropic kernels of $A[\ell]$ which intersect non trivially with one of the $\alpha_i(A[\ell^{\min(\epsilon,e_i)}])$. For $k \in \mathcal{L}$ do*
    (a) *Compute $A' = A/k$. Let $N_i' = \#\alpha_i(A'[\ell^{\min(\epsilon,e_i)}])$. If $\sum N_i' < \sum N_i$, restart the algorithm with $A = A'$ (but do not reinitialise ϵ in Step 2).*
(6) *If we get stuck and $\epsilon < \max(e_i : i = 1, 2, 3)$, set $\epsilon := \epsilon + 1$ and go back to Step 3. Otherwise return* fail*.*

3.3. **Cost of the "going-up" step.** We will see in the examples that the "going-up" step is a very important part in speeding up the CRT algorithm in practical computations. However, since it is doomed to fail in some cases (see Remark 18), we need to check that it will not dominate the complexity of the rest of the algorithm, (so that in theory there is no drawback in using it), and thus we need to estimate the cost of the "going-up" step.

The going up phase is a mix of endomorphism testing and isogeny computations. We already analysed the cost of the endomorphism testing in the preceeding section. For the isogeny computation, the points in the kernel of rational (ℓ, ℓ)-isogenies live in an extension of degree at most $\ell^2 - 1$. Transposing the analysis of Section 2.6 to this case shows that the computation of all points in these kernels takes at most $\tilde{O}(\ell^4)$. There are at most $O(\ell^3)$ such kernels, and each isogeny computation takes at most $\tilde{O}(\ell^4)$ operations in the extension. The final cost is at most $\tilde{O}(\ell^9)$ for computing all isogenies. For each of the $O(\ell^3)$ isogenous abelian varieties we do (part of) the endomorphism ring computation, which is ℓ^{2e+6} according to Section 2.6. Since the global obstruction computed is of size $O(\ell^e)$, we do at most $O(e)$ steps. The global complexity is then:

Proposition 17. *Let $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]] = \prod \ell_i^{e_i}$ be the decomposition of the index into prime factors. Then the going up phase either fails or is done in at most $\tilde{O}(\sum \ell_i^{2e_i+9})$ operations in the base field.*

Remark 18. *It is important to note that the going up phase does not always succeed. We will give some examples of that in Section 7. First, as noted in the introduction of the section, the (ℓ, ℓ)-isogeny graph is not always connected, so if we start with a curve not in the same component of a maximal curve, there is no way to find the maximal curves using only (ℓ, ℓ)-isogenies. Second, even if the curve is in the same component, finding a maximal curve may involve going through isogenous curves that increment the global obstruction, so the going up algorithm would not find it.*

In practical computations we observed the following behavior: in the very large majority of the cases where we were not able to go up, there actually did not exist any rational (ℓ, ℓ)-isogenies for any curve in the isogeny class. If χ_π is the characteristic polynomial, this can be detected by the fact that χ_π does not factor modulo ℓ as $\chi_\pi = P\overline{P} \pmod{\ell}$ (where $\overline{P}$ is the conjugate of P under the action $\pi \to p/\pi$, which sends the Frobenius to the Verschiebung). In this situation, there is no way to go up even locally at ℓ. This gives a criterion to estimate whether one can go up for this ℓ.

[1] The degree of the extension where the full ℓ^ϵ-torsion is defined depends on whether Step 1 succeeded.

4. Computing maximal curves from maximal curves

Once a maximal curve in the isogeny class has been found via the random search and "going up" steps, we use isogenies to find the other maximal curves. The set of maximal curves in the isogeny class corresponding to a fixed CM type Φ is a principal homogeneous space under the action of the Shimura class group

$$\mathfrak{C} = \{(I, \rho) \mid I \text{ a fractional } \mathcal{O}_K\text{-ideal with } I\overline{I} = \rho,\ \rho \in K^+ \text{ totally positive}\}/K^*,$$

associated to the primitive quartic CM field K, which acts by isogenies (see for instance [5, Section 3]).

However, using AVIsogenies we can only compute isogenies with a maximal isotropic kernel. In terms of the Shimura class group, the lemma below shows that this means that we can only compute the action corresponding to (equivalences classes) of elements of the form (I, ℓ) where I is an ideal in K and ℓ a prime number.

Lemma 19. *Let (I, ρ) be an element of the Shimura class group $\mathfrak{C}$ and ℓ a prime. Then the action of (I, ρ) on a maximal abelian variety A corresponds to an isogeny with maximal isotropic kernel in $A[\ell]$ if and only if $\rho = \ell$ (so if and only if I has relative norm ℓ).*

Proof. This follows from the construction of the action of $\mathfrak{C}$ on the set of maximal abelian varieties. The action is given by the isogeny $f : \mathbb{C}^2/\Lambda \to \mathbb{C}^2/I\Lambda$ and moreover the action of $\overline{I}$ corresponds to the dual isogeny $\widehat{f}$ (here we identify the abelian variety A with its dual $\widehat{A}$ via the principal polarization induced from the CM data). Since ℓ is prime, the isogeny corresponding to I is an (ℓ, ℓ) isogeny if and only if $I\overline{I} = \rho = \ell$. □

Therefore to ensure that we can find all other maximal curves using this type of isogeny we make the following heuristic assumption. Here Δ is the discriminant of K:

Assumption. *The Shimura class group $\mathfrak{C}$ is generated by elements of the form (I, ℓ) where ℓ is a polynomial in* $\log \Delta$.

Justification: We have tested this assumption on numerous examples. The assumption on the size of the isogenies will be used in the complexity analysis. At worst, we know (under GRH) that the class group of the reflex field is generated by prime ideals of degree one and of norm polynomial in $\log \Delta$ [1, Theorem 1]. (Note that the discriminant of the reflex field is $O(\Delta^2)$). But if I is such an ideal of $\mathcal{O}_{K_\Phi}$ of norm prime to p, then the element $(TN(I), N(I))$ will give a horizontal isogeny. So we will at least be able to compute all the maximal curves that are deduced from the first one by an action coming from the type norm. As we will see in the complexity analysis in Section 6, this is sufficient for most discriminants D.

Lemma 20. *Let A be an ordinary abelian surface with* $\mathrm{End}(A) \otimes \mathbb{Q} = K$*, and let $f : A \to B$ be an isogeny of degree prime to* $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$*. Then* $\mathrm{End}(A) = \mathrm{End}(B)$.

Proof. Let d be the smallest integer that factorizes through f, so $d = f\tilde{f}$. By assumption d is prime to the index. If $\alpha \in \mathrm{End}(A)$, then $f \circ \alpha \circ \tilde{f} = d\alpha$ is an endomorphism of B. Since $[\mathcal{O}_K : \mathrm{End}(B)]$ is prime to d, we have that $\alpha \in \mathrm{End}(B)$. The same argument shows that $\mathrm{End}(B) \subset \mathrm{End}(A)$, so $\mathrm{End}(A) = \mathrm{End}(B)$. □

Note that we can precompute generators of the Shimura class group since this data does not depend on the current prime p. We want to find generators of relative norm a prime $\ell \in \mathbb{Z}$ with ℓ as small as possible since this will directly influence the time spent to find the other maximal curves.

Now for a CRT prime p, there may exist among the generators we have chosen some that divide the index $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$. We can either find other generators (whose norm will be bigger), or still try to use the precomputed generators. In this case, if such a generator has norm ℓ, then not all new (ℓ, ℓ)-isogenous abelian varieties will be maximal, so we have to use the algorithm of Section 2 to test which of them is maximal. In that case, after the isogeny is applied, the ℓ^e-torsion must again be computed (with the notation of Section 3), along with the action of the generators of $(\mathcal{O}_K)_\ell$ over $\mathbb{Z}[\pi, \overline{\pi}]_\ell$. The trade-off depends then on the degree of the extension field required to compute the ℓ^e-torsion for small ℓ dividing the index versus the degree of the field of definition for the points in the kernel of the ℓ-isogeny for ℓ not dividing the index.

Finally, we can also use the group structure of the Shimura class group as follows: suppose that we have computed maximal curves corresponding to the action of $\alpha_1, \dots, \alpha_t \in \mathfrak{C}$, and we want to find new maximal curves by computing (ℓ, ℓ)-isogeny graphs starting from these curves. Then if $\mathfrak{C}(\ell)$ is the set of elements of the form (I, ℓ) in $\mathfrak{C}$, then the number of maximal curves that we can find in this way is the cardinality of the subgroup generated by the α_i and $\mathfrak{C}(\ell)$. In particular, as soon as we reach this number, we can stop the computation since it will not yield any new maximal curves. This is particularly useful when ℓ divides the index since it avoids some endomorphism tests. In the isogeny graph computation done by AVIsogenies, each node is computed twice since there are two edges between adjacent nodes (corresponding to the isogeny and the dual). Here since we know the number of nodes, we can abort the computation early.

We thus obtain the following algorithm:

Algorithm 21. *Finding all maximal curves from one maximal curve.*

Input: *An ordinary abelian variety $A/\mathbb{F}_p$ with CM by (O_K, Φ).*
Output: *All abelian varieties over $\mathbb{F}_p$ with CM by (O_K, Φ).*

Precompute *a set of generators of the Shimura class group with relative norm ℓ as small as possible (the set is not chosen to be minimal, on the contrary we want some redundancy). For each of the generators, compute the extension degree of the field of definition of the geometric points of the kernel corresponding to this generator.*

(1) *For each generator of (relative) norm ℓ dividing the index, replace the previous degree by the degree of the extension where the ℓ^e-torsion lives (usually e is the ℓ-valuation of the index, but the tricks from Section 2 can sometimes reduce it).*
(2) *Sort the generators according to the corresponding degrees to get a list $(g_1, \dots, g_n)$.*
(3) *For each generator g_i on the list, let ℓ_i be its norm and do*
    (a) *Compute the varieties (ℓ_i, ℓ_i)-isogenous to the one already found. If ℓ_i divides the index, then do an endomorphism ring computation from Section 2 and keep only the maximal curves*
    (b) *Repeat until the number of (maximal) abelian varieties is $\#\langle \mathfrak{C}(\ell_1), \dots, \mathfrak{C}(\ell_i) \rangle$.*

5. The CRT Step

Contrary to the elliptic curve case, the coefficients of the class polynomials in genus 2 are rational numbers, not integers. We estimate the denominators using the Bruinier-Yang conjectural formula [6] (proved only for special cases [31, 32]), with minor adjustments from [17], and adapting it to the fact that we use invariants from [27, Appendix 3], which alters the denominator formulas by small powers of 2. A formula for the factorization of the denominators which holds for general primitive quartic CM fields was recently given in [19] which produces a multiple of the denominators when allowing for cancellation with the numerators and the case where K^+ does not have class number 1. As in [12, Theorem 3], we can multiply by the denominators and then use the CRT to reconstruct the polynomials.

5.1. **Sieving the CRT primes.** To determine whether to use a CRT prime in the CRT algorithm, we check if the corresponding isogeny class is large enough. There are p^3 isomorphism classes of genus 2 curves over $\mathbb{F}_p$, and since the area of Figure 10.1 in [21] is $32/3$, there are approximately $32/3p^{3/2}$ isogeny classes. We keep p if the size of the isogeny class is of size roughly $p^{3/2}$. We estimate the size of this isogeny class by using Lemma 6.3 in [21] for each order (stable by conjugation) between $\mathbb{Z}[\pi, \overline{\pi}]$ and O_K.

In practice, we are only interested in the number of curves from which we can go up. This is harder to estimate, but numerous computations showed that the main obstruction to going up occurs when there are no (ℓ, ℓ)-isogenies with rational kernel at all. But this case is easy to detect (see [3]). So in the previous estimate, we discount the orders whose index is divisible by such an ℓ.

Finally, we use a dynamic approach for the prime selection: we use a prime if the probability of finding a maximal curve with the going up algorithm is better than a certain threshold (depending on the size of the prime), but we go back to previously discarded (smaller) primes if they satisfy the threshold for the current size of primes we are considering.

5.2. **The CRT.** In the cyclic case, we compute the class polynomials modulo small integer primes, and we use the CRT to get the result modulo the product P (the "precision") of these small primes. Once the precision is enough, we can recover the polynomials modulo $\mathbb{Z}$, by lifting each coefficient to $\mathbb{Z}$ in the interval $[-P/2, P/2]$.

In the dihedral case, the primes are in $\mathcal{O}_{K^+_\Phi}$, and so is the precision ideal P. Here we explain how to lift a coefficient $x \bmod P$ to $\mathcal{O}_{K^+_\Phi}$. Take the Minkowski embedding of a lift of x, and find the closest vector c_x in the lattice associated to P in the Minkowski embedding. Then c_x corresponds to an element of the ideal P, and our final lift is $x - c_x$. We note that the lattice is of rank 2, so we can directly compute the closest vector rather than doing an LLL approximation.

5.3. **Lifting without denominators.** We note that in the dihedral case, the denominator from the formulas in [6, 17, 19] is too large, as it takes into account both CM types. This increases the size of the coefficients we compute, so that using those denominator formulas does not actually give better results than doing a rational reconstruction directly.

With the notation from above, from $x \bmod P$ we want to do a rational lift of x. This time we embed the lattice associated to P into the lattice of rank 3 obtained by adjoining the vector $[Cx_1, Cx_2, C]$ where x_1 and x_2 are the two real embeddings of (a lift of) x and C is a constant accounting for how skewed we expect the size of the denominator to be compared to the numerators. A minimal vector in this lattice will correspond to an element $N = c + Dx$ where $c \in p$ and D is an integer. We then take N/D as our lift for x.

This solution requires the precision to be the sum of the bit sizes of the numerators and denominator, so it can be even better than using the denominator formulas for small denominators where there may be cancellation with the numerators.

6. Complexity

In this section, we analyze the effect of the going up algorithm of Section 3 (which we will also call the vertical step since it deals with abelian varieties which have endomorphism rings which are different orders in K) and the effect of finding all other maximal curves from one maximal curve from Section 4 (the horizontal step) to the asymptotic complexity. Most of the discussion is heuristic.

We begin with a quick reminder of the rough complexity analysis of the CRT method in the elliptic curve case, where K is a quadratic imaginary field. There is only one class polynomial H, whose degree is the class number of $\mathcal{O}_K$, and classical bounds give that $\deg H = O(\sqrt{\Delta})$ where Δ is the discriminant of $\mathcal{O}_K$. Likewise, the coefficients of H have size $\tilde{O}(\sqrt{\Delta})$. So the whole class polynomial is of size $\tilde{O}(\Delta)$.

Each CRT prime p gives $\log(p)$ bits of information, so neglecting logarithmic factors, we need $\sqrt{\Delta}$ primes. CRT primes split completely in the Hilbert class field of K, whose Galois group is $\mathrm{Cl}(\mathcal{O}_K)$, so by the Cebotarev theorem the density of CRT primes is roughly $1/\#\,\mathrm{Cl}(\mathcal{O}_K) \simeq 1/\Delta$. Neglecting logarithmic factors again, the biggest prime p is of size $\tilde{O}(\Delta)$.

Now there are $O(p)$ isomorphism classes of elliptic curves, and $\tilde{O}(\sqrt{\Delta})$ maximal curves, so one is found in time $\tilde{O}(p/\sqrt{\Delta}) = \tilde{O}(\sqrt{p})$. Once one maximal curve is found, all others can be obtained using isogenies of degree logarithmic in Δ, so one can recover all maximal elliptic curves over $\mathbb{F}_p$ in time $\tilde{O}(\sqrt{p}) = \tilde{O}(\sqrt{\Delta})$.

We need $\sqrt{\Delta}$ CRT primes, so the total cost is $\tilde{O}(\Delta)$. The CRT reconstruction can be done in quasi-linear time too, so in the end the algorithm is quasi-linear, even without using a vertical step. Not using the horizontal step gives a complexity of $\tilde{O}(\Delta^{3/2})$.

In genus 2, let $\Delta_0 = \Delta_{K^+/\mathbb{Q}}$ and $\Delta_1 = N_{K^+/\mathbb{Q}}(\Delta_{K/K^+})$, so that $\Delta = \Delta_{K/\mathbb{Q}} = \Delta_1\Delta_0^2$. Then the degree of the class polynomials is $\tilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$ while the height of their coefficients is bounded by $\tilde{O}(\Delta_0^{5/2}\Delta_1^{3/2})$ ([27, Section II.9], [16]). In practice, we observe [27, Appendix 3] that they are bounded by $\tilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$ and we will use this bound in the following. According to [5, Section 6.4], the smallest prime is of size $\tilde{O}(\Delta_0\Delta_1)$. We need $\tilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$ CRT primes, and an analysis using [18] as in [2, Lemma 5.3] show that the largest prime is also $\tilde{O}(\Delta_0\Delta_1)$. We remark that the sieving phase does not affect the size of the largest prime (apart from the constant in the big O) as long as we sieve a positive density of CRT primes.

For the horizontal step, the isogeny computation involves primes of size logarithmic in Δ, so the cost of this step is quasi-linear in the number $\tilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$ of maximal curves. This is under the assumption from Section 4. Without this assumption, what we know is that for each ideal I in $\mathcal{O}_{K_\Phi}$ of norm prime to p, the element $(TN(I), N(I))$ is an element of the Shimura class group whose action is given by a maximally isotropic kernel. In the horizontal step, we can then compute the action of $TN(\mathrm{Cl}(\mathcal{O}_{K_\Phi}))$ by isogenies of size logarithmic in Δ. By Lemma 6.5 of [5], the cofactor is bounded by $2^{6w(D)+1}$, where $w(D)$ is the number of prime divisors of D. This gives a bound on the number of horizontal isogeny steps we need to take. As remarked in [5], outside a zero-density subset of very smooth integers, $w(n) < 2\log\log n$ so the corresponding factor can be absorbed into the $\tilde{O}$-notation.

On the contrary, the complexity of the endomorphism ring computation and the going up phase involves the largest prime power dividing the index $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$. According to Proposition 6.1 of [14] we have that $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]] \leq \frac{16p^2}{\sqrt{\Delta}}$. For the size of the CRT prime we are considering, we see that $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]] = \tilde{O}(\Delta_0\Delta_1^{3/2})$. We fix $\epsilon = 1/2$. Assuming that the index is uniformly distributed, [11] showed that there is a positive density of CRT primes where the largest prime power dividing the index is $O(\Delta_0^{\epsilon/100}\Delta_1^{\epsilon/100})$. By the complexity analysis of Sections 2.6 and 3.3, we see then that there is a positive density of primes where these algorithms take time at most $O(\Delta_0^\epsilon\Delta_1^\epsilon)$.

We then let $p = \tilde{O}(\Delta_0\Delta_1)$ be a CRT prime. There are $O(\sqrt{p})$ maximal curves, so we expect the isogeny class to be of size $\Theta(p^{3/2})$ (see Heuristic 6.6 in [5]), and there are p^3 isomorphism classes of curves. The original CRT algorithm of [12, 14] looped through all p^3 curves and tested if the endomorphism ring is maximal. This takes $\tilde{O}(\Delta_0^3\Delta_1^3 + O(\Delta_0^{3/2+\epsilon}\Delta_1^{3/2+\epsilon})$ per CRT prime, for a total cost (since $\tilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$ CRT primes are needed) of $\tilde{O}(\Delta_0^{7/2}\Delta_1^{7/2})$ with our choice of ϵ.

The approach of [5] is to find only one of them and use horizontal isogenies to find the others. With the improvements proposed in this paper (using all horizontal isogenies and not only those coming from the type norm, and the improved endomorphism ring computation): we find a cost of $\tilde{O}(\Delta_0^{5/2}\Delta_1^{5/2}) + O(\Delta_0^{3/2+\epsilon}\Delta_1^{3/2+\epsilon})$ per CRT prime. The total cost is then $\tilde{O}(\Delta_0^3\Delta_1^3)$.

With our method, we need to find a curve in the isogeny class where the going up step yields a maximal curve. Finding a curve in the isogeny class takes time $O(p^{3/2})$. If X is the number of going up steps we need to try on average, the cost per CRT prime is then $\tilde{O}(X(\Delta_0^{3/2}\Delta_1^{3/2} + \Delta_0^\epsilon\Delta_1^\epsilon))$. At best, $X = O(1)$, and we have a total cost of $\tilde{O}(\Delta_0^2\Delta_1^2)$ from CRT primes. So at best we have a quasi-quadratic complexity, while the CRT itself is quasi-linear, so is negligible. We see that we are still far from quasi-linearity achieved by the analytic method. At worst, $X = O(p)$ (number of random tries in the isogeny class until we find a maximal one directly), and we recover the quasi-cubic complexity of the previous method.

To improve the complexity, there are two possibilities: the first is to increase the probability of success of the going up method. This requires an algorithm to compute isogenies with cyclic kernels. But even with that, we achieve at most quasi-quadratic complexity because the size of the isogeny class is too small compared to the size of the search space. This is the case because the algorithm computes the class polynomials (a scheme of dimension 0) directly from the moduli space of dimension 3 of all abelian surfaces. In contrast, in the elliptic curve case, the algorithm searches a space of dimension 1 for elements of a space of dimension 0. It would be interesting to find convenient subspaces of the moduli space of smaller dimension, and to work over them. One example would be to use Humbert surfaces, which are of dimension 2, and Gundlach invariants, as proposed in [20].

7. Examples

7.1. **Improvements due to the going up phase.**

We first look at improvements due solely to the going up phase. The new timings for the case $K = \mathbb{Q}(i\sqrt{29 + 2\sqrt{29}})$, $\mathfrak{C}(O_K) = \{0\}$, are given in the next table, compared to old timings from [14, Section 9]. This is a cyclic Galois example with class number one, so there is only one maximal curve and the algorithm from Section 4 is not used.

p	l^d	α_d	# Curves	Estimate	Time (old)	Time (new)
7	-	-	1	1	0.3	$0 + 0.1$
23	$\mathbf{13}$	84	15	2 (16)	$9 + 70.7$	$0.4 + 24.6$
53	7	3	7	7	$105 + 0.5$	$7.7 + 0.5$
59	$2, \mathbf{5}$	$1, 12$	322	48 (286)	$164 + 6.4$	$1.4 + 0.6$
83	$3, 5$	$4, 24$	77	108	$431 + 9.8$	$2.4 + 1.1$
103	$\mathit{67}$	$\mathit{1122}$	-	-	-	-
107	$7, \mathbf{13}$	$3, 21$	105	8 (107)	$963 + 69.3$	-
139	$\mathbf{5}^2, 7$	$60, 2$	259	9 (260)	$2189 + 62.1$	-
181	3	1	161	135	$5040 + 3.6$	$4.5 + 0.2$
197	$5, 109$	$24, \mathit{5940}$	-	-	-	-
199	$\mathbf{5}^2$	60	37	2 (39)	$10440 + 35.1$	-
223	$2, \mathit{23}$	$1, 11$	1058	39 (914)	$10440 + 35.1$	-
227	109	$\mathit{1485}$	-	-	-	-
233	$5, 7, \mathbf{13}$	$8, 3, 28$	735	55 (770)	$11580 + 141.6$	$88.3 + 29.4$
239	$7, 109$	$6, \mathit{297}$	-	-	-	-
257	$3, 7, \mathbf{13}$	$4, 6, 84$	1155	109 (1521)	$17160 + 382.8$	-
313	$3, \mathbf{13}$	$1, 14$	?	146 (2035)	-	$165 + 14.7$
373	$5, 7$	$6, 24$	?	312	-	$183.4 + 3.8$
541	$2, 7, \mathbf{13}$	$1, 3, 14$	?	294 (4106)	-	$91 + 5.5$
571	$3, \mathbf{5}, 7$	$2, 6, 6$	?	1111 (6663)	-	$96.6 + 3.1$
					56585s	776s

The first column indicates the CRT prime used. The second one the ℓ^d-torsion subgroups required to compute whether a curve is maximal; the ℓ is in bold if there is not any rational (ℓ, ℓ)-isogeny, so we can't go up for this ℓ (and it is in italic if there exist (ℓ, ℓ)-isogenies, but they are too expensive to compute). The third column gives the corresponding degree where the points of these subgroups live (this degree is in italic when it is too high, computing the ℓ^d-torsion would be too expensive). The fourth column indicates the total number of curves in the isogeny class (this is found via the algorithm from [14] which goes though all the curves), while the fifth gives an estimate of the number of curves from which we can go up (and the number in parenthesis is our estimate of the total number of curves in the isogeny class). The last two columns give the timings of the old and new algorithms, split into "Time exploring curves" + "Time spent computing endomorphism rings/Time spent going up".

Note that much less time is spent exploring curves with the new algorithm due to the going up algorithm. Also note that, even though the going up phase is more complicated, it is still less costly than the computation of the endomorphism rings in the old algorithm, due to the improvements described in Section 2 and the fact that the new version calls it less often.

The trade-offs in the going up step depend on the discriminant of the CM field K. The more CRT primes we need, the bigger the isogenies and the bigger the degrees in the endomorphism ring computations we allow. Note that computing (ℓ, ℓ)-isogenies requires $\tilde{O}(\ell^2)$ operations in the field where the points of the kernel are defined when ℓ is congruent to $1 \pmod 4$, but $\tilde{O}(\ell^4)$ when ℓ is congruent to 3 (mod 4). So in the above example, we computed the $(109, 109)$-isogenies faster than the $(23, 23)$-isogenies.

7.2. **Dihedral examples.** Here we illustrate our new CRT algorithm for dihedral fields, for $K = \mathbb{Q}(X)/(X^4 + 13X^2 + 41)$ with $\mathfrak{C}(K) \simeq \{0\}$.

We first compute the class polynomials over $\mathbb{Z}$ using Spallek's invariants, and obtain the following polynomials in 5956 seconds: $H_1 = 64X^2 + 14761305216X - 111577100832000000$, $H_2 = 16X^2 + 72590904X - 8609344200000$, $H_3 = 16X^2 + 28820286X - 303718531500$.

Next we compute them over the real subfield and use the invariants from [27, Appendix 3]. We get $H_1 = 256X - 2030994 + 56133\alpha$, $H_2 = 128X + 12637944 - 2224908\alpha$, $H_3 = 65536X - 11920680322632 + 1305660546324\alpha$ where α is a root of $X^2 - 3534X + 177505$, so that $O_{K_0^+} = \mathbb{Z}[\alpha]$. This computation took 1401 seconds, so in this case, the speedup due to using better invariants and computing over the real subfield is more than 4-fold.

References

[1] E. Bach *Explicit Bounds for Primality Testing and Related Problems*, Mathematics of Computations, **55**, No. 191, (1990) 335–380.

[2] J. Belding, R. Bröker, A. Enge, K. Lauter, *Computing Hilbert class polynomials*, ANTS VIII, Springer LNCS **5011** (2008), 282–295.

[3] G. Bisson, R. Cosset, D. Robert, *AVIsogenies (Abelian Varieties and Isogenies)*, Magma package for explicit isogenies between abelian varieties, `http://avisogenies.gforge.inria.fr`.

[4] G. Bisson, *Endomorphism Rings in Cryptography*, PhD Thesis (2011).

[5] R. Bröker, D. Gruenewald, K. Lauter, *Explicit CM-theory for level 2-structures on abelian surfaces*, Algebra and Number Theory, Vol. 5 (2011), No. 4, 495–528. DOI: 10.2140/ant.2011.5.495.

[6] J. Bruinier, T. Yang, CM-values of Hilbert modular functions, Invent. Math., **163**, no. 2, (2006) 229–288.

[7] Robert Carls, David Kohel, David Lubicz, *Higher dimensional 3-adic CM construction*, J. of Algebra, 319-4 (2008), 971–2006.

[8] Robert Carls, David Lubicz, *A p-adic quasi-quadratic point counting algorithm*, *Int. Math. Res. Not.*, 2008.

[9] R. Cosset, D. Robert, *An algorithm for computing (ℓ, ℓ)-isogenies in polynomial time on Jacobians of hyperelliptic curves of genus* 2, Preprint 2011, `http://eprint.iacr.org/2011/143`.

[10] J.M. Couveignes, *Linearizing torsion classes in the Picard group of algebraic curves over finite fields*, Journal of Algebra, **321**, no. 8, (2009) 2085–2118.

[11] K. Dickman, *On the frequency of numbers containing prime factors of a certain relative magnitude*, Ark. Mat. Astr. Fys. 22A (1930), 1–14.

[12] K. Eisenträger, K. Lauter, *A CRT algorithm for constructing genus 2 curves over finite fields*, In: Arithmetic, Geometry and Coding Theory AGCT-10 (2005), Séminaires et Congrès **21**, Société Mathématique de France (2009) 161–176. `http://arxiv.org/abs/math.NT/0405305`

[13] A. Enge, A.V. Sutherland, *Class invariants by the CRT method*, ANTS IX: Proceedings of the Algorithmic Number Theory 9th International Symposium, Nancy (2010), 142–156.

[14] D. Freeman, K. Lauter, *Computing endomorphism rings of Jacobians of genus 2 curves over finite fields*, in Algebraic Geometry and its Applications, World Scientific (2008), 29–66.

[15] P. Gaudry, T. Houtmann, D. Kohel, C. Ritzenthaler, and A. Weng, *The 2-adic CM method for genus 2 curves with application to cryptography*, Asiacrypt 2006 (Shanghai) Lect. Notes in Comp. Sci., 4284, 114–129, Springer-Verlag, 2006.

[16] E. Goren, K. Lauter, *Genus 2 curves with complex multiplication*, International Mathematics Research Notices (2011), 75 pp. doi: 10.1093/imrn/rnr052.

[17] H. Grundman, J. Johnson-Leung, K. Lauter, A. Salerno, B. Viray, E. Wittenborn, *Igusa class polynomials, embeddings of quartic CM fields, and arithmetic intersection theory*, WIN-Women in Numbers: Research Directions in Number Theory, Fields Institute Communications Series, Volume 60 (2011) 35–61.

[18] J. C. Lagarias, A. M. Odlyzko, *Effective versions of the Chebotarev density theorem*, Algebraic number fields: L-functions and Galois properties (Proc. Sympos., Univ. Durham, Duram, 1975), Academic Press, 1977, pp. 409–464.

[19] K. Lauter, B. Viray, *An arithmetic intersection formula for denominators of Igusa class polynomials*, Preprint, 2011.

[20] K. Lauter, T. Yang, *Computing genus 2 curves from invariants on the Hilbert moduli space*, Journal of Number Theory, Elliptic Curve Cryptography Volume **131**, Issue 5 (2011), 936–958.

[21] H. W. Lenstra, Jr., J. Pila, C. Pomerance, *A hyperelliptic smoothness test II*, Proc. London Math. Soc., (3) **84** (2002), 105–146.

[22] D. Lubicz, D. Robert, *Computing isogenies between abelian varieties*, to appear in Compositio Mathematica, `http://arxiv.org/pdf/1001.2016`.

[23] J.-F. Mestre, *Construction de courbes de genre 2 à partir de leurs modules*, in Effective methods in algebraic geometry, Birkhäuser Progr. Math. **94** (1991), 313–334.

[24] D. Robert, *Theta functions and applications in cryptography*, PhD Thesis, Université Henri-Poincarré, (2010).

[25] G. Shimura, *Abelian varieties with complex multiplication and modular functions*, Princeton Mathematical Series **46**, Princeton University Press New Jersey, 1998.

[26] A.-M. Spallek, *Kurven vom Geschlecht 2 und ihre Anwendung in Public-Key-Kryptosystemen*, PhD thesis, Universität Gesamthochschule Essen, 1994.

[27] M. Streng, *Complex multiplication of abelian surfaces*, PhD-thesis, Universiteit Leiden, 2010.

[28] A. Sutherland, *Computing Hilbert class polynomials with the Chinese Remainder Theorem*, Mathematics of Computation **80** (2011), pp. 501-538.

[29] P. van Wamelen. *Examples of genus two CM curves defined over the rationals*, Math. Comp., 68 (1999), 307–320.

[30] A. Weng, *Constructing hyperelliptic curves of genus 2 suitable for cryptography*, Math. Comp. 72 (2003), 435–458.

[31] T. Yang, *An arithmetic intersection formula on Hilbert modular surfaces*, *Amer. J. Math.*, 132 (2010), 1275–1309.

[32] T. Yang, *Arithmetic intersection on a Hilbert modular surface and the Faltings height*, Preprint, 2007.

Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA.
E-mail address: `damien.robert@microsoft.com`

Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA.
E-mail address: `klauter@microsoft.com`